MTHRTL

```
MM      MM  TTTTTTTTTT  HH      HH  CCCCCCCC   GGGGGGGG   SSSSSSSS   QQQQQQ    RRRRRRRR   TTTTTTTTTT
MM      MM  TTTTTTTTTT  HH      HH  CCCCCCCC   GGGGGGGG   SSSSSSSS   QQQQQQ    RRRRRRRR   TTTTTTTTTT
MMMM  MMMM      TT      HH      HH  CC         GG             SS    QQ    QQ   RR      RR     TT
MMMM  MMMM      TT      HH      HH  CC         GG             SS    QQ    QQ   RR      RR     TT
MM  MM  MM      TT      HH      HH  CC         GG             SS    QQ    QQ   RR      RR     TT
MM  MM  MM      TT      HH      HH  CC         GG             SS    QQ    QQ   RR      RR     TT
MM      MM      TT      HHHHHHHHHH  CC         GG                   QQ    QQ   RRRRRRRR       TT
MM      MM      TT      HHHHHHHHHH  CC         GG          SSSSSS   QQ    QQ   RRRRRRRR       TT
MM      MM      TT      HH      HH  CC         GG  GGGGG   SSSSSS   QQ QQ QQ   RR  RR         TT
MM      MM      TT      HH      HH  CC         GG  GGGGGG      SS   QQ QQ QQ   RR  RR         TT
MM      MM      TT      HH      HH  CC         GG     GG       SS   QQ    QQ   RR      RR     TT
MM      MM      TT      HH      HH  CC         GG     GG       SS   QQ    QQ   RR      RR     TT
MM      MM      TT      HH      HH  CCCCCCCC   GGGGGG     SSSSSSSS  QQQQ  QQ   RR      RR     TT
MM      MM      TT      HH      HH  CCCCCCCC   GGGGGG     SSSSSSSS  QQQQ  QQ   RR      RR     TT
```

```
LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII      SSSSSSSS
LLLLLLLLLL      IIIIII      SSSSSSSS
```

```
0000    1              .TITLE  MTH$CGSQRT
0000    2              .IDENT  /1-002/         ; File: MTHCSQRT.MAR  SBL1002
0000    3
0000    4
0000    5     ;*********************************************************************
0000    6     ;*                                                                   *
0000    7     ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
0000    8     ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
0000    9     ;*  ALL RIGHTS RESERVED.                                             *
0000   10     ;*                                                                   *
0000   11     ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000   12     ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0000   13     ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0000   14     ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000   15     ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0000   16     ;*  TRANSFERRED.                                                      *
0000   17     ;*                                                                   *
0000   18     ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0000   19     ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0000   20     ;*  CORPORATION.                                                      *
0000   21     ;*                                                                   *
0000   22     ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0000   23     ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.          *
0000   24     ;*                                                                   *
0000   25     ;*                                                                   *
0000   26     ;*********************************************************************
0000   27
0000   28     ;
0000   29     ; FACILITY: MATH LIBRARY
0000   30     ;++
0000   31     ; ABSTRACT:
0000   32     ;       This module contains routine MTH$CGSQRT - compute
0000   33     ;       G COMPLEX*16 square root.
0000   34     ;
0000   35     ;--
0000   36     ;
0000   37     ; VERSION: 1
0000   38     ;
0000   39     ; HISTORY:
0000   40     ;
0000   41     ; AUTHOR:
0000   42     ;       Steven B. Lionel, 24-July-1979
0000   43     ;
0000   44     ; MODIFIED BY:
0000   45     ;
0000   46     ;
0000   47     ;
```

MTH$CGSQRT
1-002

M 5

HISTORY ; Detailed Current Edit History 16-SEP-1984 01:10:12 VAX/VMS Macro V04-00 Page 2
                                         6-SEP-1984 11:21:11 [MTHRTL.SRC]MTHCGSQRT.MAR;1 (2)

```
0000    49              .SBTTL  HISTORY          ; Detailed Current Edit History
0000    50
0000    51
0000    52  ; Edit History
0000    53  ;
0000    54  ; 1-001 - Adapted from MTH$CSQRT version 1-003.  SBL 24-July-1979
0000    55  ; 1-002 - Use general mode addressing.  SBL 30-Nov-1981
```

```
0000          57              .SBTTL  DECLARATIONS
0000          58
0000          59      ;
0000          60      ; INCLUDE FILES:
0000          61      ;
0000          62
0000          63      ;
0000          64      ; EXTERNAL SYMBOLS:
0000          65              .DSABL  GBL
0000          66              .EXTRN  MTH$CGABS
0000          67              .EXTRN  MTH$GSQRT_R5
0000          68
0000          69      ;
0000          70      ; MACROS:
0000          71      ;       NONE
0000          72
0000          73      ;
0000          74      ; PSECT DECLARATIONS:
00000000      75              .PSECT  _MTH$CODE        PIC, SHR, LONG, EXE, NOWRT
0000          76
0000          77      ;
0000          78      ; EQUATED SYMBOLS:
0000          79      ;
0000          80      ;
0000          81      ; OWN STORAGE:
0000          82      ;       NONE
```

MTH$CGSQRT
1-002

B 6

16-SEP-1984 01:10:12   VAX/VMS Macro V04-00   Page   4
MTH$CGSQRT - compute G COMPLEX*16 square   6-SEP-1984 11:21:11   [MTHRTL.SRC]MTHCGSQRT.MAR;1   (4)

```
                        0000      84          .SBTTL  MTH$CGSQRT - compute G COMPLEX*16 square root
                        0000      85
                        0000      86  ;++
                        0000      87  ; FUNCTIONAL DESCRIPTION:
                        0000      88  ;
                        0000      89  ;       The square root of a complex number (r, i) is computed
                        0000      90  ;       as follows:
                        0000      91  ;
                        0000      92  ;       ROOT = SQRT((ABS(r) + CABS((r, i))) / 2)
                        0000      93  ;       Q = i / (2*ROOT)
                        0000      94  ;
                        0000      95  ;
                        0000      96  ;       r          i          CSQRT((r, i))
                        0000      97  ;       -          -          -------------
                        0000      98  ;
                        0000      99  ;       >=0        any        (ROOT, Q)
                        0000     100  ;       <0         >=0        (Q, ROOT)
                        0000     101  ;       <0         <0         (-Q, -ROOT)
                        0000     102  ;
                        0000     103  ; CALLING SEQUENCE:
                        0000     104  ;
                        0000     105  ;       CALL MTH$CGSQRT (result.wgc.r, arg.rgc.r)
                        0000     106  ;
                        0000     107  ; INPUT PARAMETERS:
                        0000     108  ;
             00000008   0000     109  ;       arg     = 8                     ; The G COMPLEX*16 argument, passed
                        0000     110  ;                                       ; by reference.
                        0000     111  ;
                        0000     112  ; IMPLICIT INPUTS:
                        0000     113  ;       NONE
                        0000     114  ;
                        0000     115  ; OUTPUT PARAMETERS:
                        0000     116  ;
             00000004   0000     117  ;       result  = 4                     ; The G COMPLEX*16 result, passed by
                        0000     118  ;                                       ; reference.
                        0000     119  ;
                        0000     120  ; IMPLICIT OUTPUTS:
                        0000     121  ;       NONE
                        0000     122  ;
                        0000     123  ; COMPLETION CODES:
                        0000     124  ;       NONE
                        0000     125  ;
                        0000     126  ; SIDE EFFECTS:
                        0000     127  ;
                        0000     128  ;       SS$_ROPRAND     If either part of argument is reserved operand.
                        0000     129  ;
                        0000     130  ;--
                        0000     131
                        0000     132
                 007C   0000     133          .ENTRY  MTH$CGSQRT,     ^M<R2, R3, R4, R5, R6>
                        0002     134          MTH$FLAG_JACKET                 ; flag as math routine
6D  00000000'GF   9E    0002              MOVAB   G^MTH$$JACKET_HND, (FP)
                        0009                                              ; set handler address to jacket
                        0009                                              ; handler
                        0009
                        0009     135
```

```
          52   08 BC 50FD  0009  136           MOVG    aarg(AP), R2        ; R2-R3 = r
        52 8000 8F   AA    000E  137           BICW    #^X8000, R2         ; R2-R3 = ABS(r)
            08 AC   DD      0013  138           PUSHL   arg(AP)             ; Put address of arg on stack
   00000000'GF   01  FB     0016  139           CALLS   #1, G^MTH$CGABS     ; R0-R1 = CABS((r, i))
            50   52 40FD    001D  140           ADDG2   R2, R0              ; R0-R1 = ABS(r) + CABS((r, i))
            50   00 44FD    0021  141           MULG2   #0.5, R0            ; R0-R1 = (ABS(r) + CABS((r, i))) / 2
   00000000'GF     16       0025  142           JSB     G^MTH$GSQRT_R5      ; R0-R1 = ROOT = SQRT(above)
          52   08 AC   D0   002B  143           MOVL    arg(AP), R2         ; R2 -> (r, i)
            50   53FD       002F  144           TSTG    R0                  ; is ROOT zero?
            04   12         0032  145           BNEQ    1$                  ; no, go ahead
            55   7C         0034  146           CLRG    R5                  ; make zero quotient
            0A   11         0036  147           BRB     2$                  ; skip divide
     55   08 A2 50 47FD     0038  148  1$:      DIVG3   R0, 8(R2), R5       ; R5 = i / ROOT
            55   00 44FD    003E  149           MULG2   #0.5, R5            ; R5 = Q = i / (2 * ROOT)
            82   53FD       0042  150  2$:      TSTG    (R2)+               ; if r positive,
            18   18         0045  151           BGEQ    RETRN               ; then return (ROOT, Q)
          53   50   7D      0047  152           MOVQ    R0, R3              ; else switch ROOT and Q
            62   53FD       004A  153           TSTG    (R2)                ; if i positive
            0A   18         004D  154           BGEQ    RETRN1              ; then return (Q, ROOT)
            50   55 52FD    004F  155           MNEGG   R5, R0              ; else negate ROOT anG Q
            55   53 52FD    0053  156           MNEGG   R3, R5              ; and return (-Q, -ROOT)
            06   11         0057  157           BRB     RETRN
                           0059  158
                           0059  159  RETRN1:
          50   55   7D      0059  160           MOVQ    R5, R0              ; continue to swap ROOT and Q
          55   53   7D      005C  161           MOVQ    R3, R5              ; and return (Q, ROOT)
                           005F  162  RETRN:
          52   04 AC   D0   005F  163           MOVL    result(AP), R2      ; result address
            82   50   7D    0063  164           MOVQ    R0, (R2)+           ; real part
            62   55   7D    0066  165           MOVQ    R5, (R2)            ; imaginary part
                  04        0069  166           RET
                           006A  167
                           006A  168
                           006A  169           .END
```

```
ARG                  = 00000008
MTH$$JACKET_HND      ********   X   01
MTH$CGABS            ********   X   00
MTH$CGSQRT           00000000 RG    01
MTH$GSQRT_R5         ********   X   00
RCSULT               = 00000004
RETRN                0000005F R     01
RETRN1               00000059 R     01
```

+-------------------+
! Psect synopsis !
+-------------------+

PSECT name	Allocation		PSECT No.	Attributes										
. ABS .	00000000 (	0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE
_MTH$CODE	0000006A (	106.)	01 (1.)	PIC	USR	CON	REL	LCL	SHR	EXE	RD	NOWRT	NOVEC	LONG

+-----------------------------+
! Performance indicators !
+-----------------------------+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.08	00:00:00.58
Command processing	122	00:00:00.64	00:00:04.27
Pass 1	89	00:00:00.72	00:00:02.57
Symbol table sort	0	00:00:00.00	00:00:00.00
Pass 2	45	00:00:00.50	00:00:02.14
Symbol table output	2	00:00:00.02	00:00:00.23
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	299	00:00:01.98	00:00:09.81

The working set limit was 900 pages.
2877 bytes (6 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 8 non-local and 2 local symbols.
229 source lines were read in Pass 1, producing 11 object records in Pass 2.
1 page of virtual memory was used to define 1 macro.

+-------------------------------+
! Macro library statistics !
+-------------------------------+

Macro library name	Macros defined
_255DUA28:[SYSLIB]STARLET.MLB;2	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:MTHCGSQRT/OBJ=OBJ$:MTHCGSQRT MSRC$:MTHJACKET/UPDATE=(ENH$:MTHJACKET)+MS

MTHCVTDG
.LIS

MTHDACOS
.LIS

MTHCGABS
.LIS

MTHCDSINC
.LIS

MTHCLOG
.LIS

MTHDASIN
.LIS

MTHCGLOG
.LIS

MTHCONVER
.LIS

MTHDATAN
.LIS

MTHCSQRT
.LIS

MTHCEXP
.LIS

MTHCGSQRT
.LIS

MTHCGEXP.
.LIS

MTHCSINCO
.LIS

MTHCONJG
.LIS

MTHCDSQRT
.LIS

MTHCGSINC
.LIS

MTHCOSH
.LIS

MTHDACOS
.LIS